

Contributing to LIGO GW Detection

Development Setup

```
# Install in development mode  
pip install -e ".[dev]"  
  
# Run tests  
pytest tests/ -v  
  
# Format code  
black ligo_gw/ tests/  
  
# Lint  
flake8 ligo_gw/
```

Code Style

- Follow PEP 8
- Use type hints where possible
- Docstring format: Google style
- Max line length: 88 characters

Week 1 Task Assignment

Suggested task breakdown for parallel development:

Data Loaders (2 days)

- **Owner:** Implement `ligo_gw/data/loaders.py`
- Parse FITS files (use `astropy.io.fits`)
- Whitening algorithm (bandpass + PSD normalization)
- Normalization and windowing
- **Test:** `tests/test_data.py` should pass

Wavelet Transforms (1.5 days)

- **Owner:** Implement `ligo_gw/data/transforms.py`
- CWT using `scipy.signal.morlet2`
- STFT using `scipy.signal.spectrogram`
- dB scaling and visualization
- **Test:** `tests/test_data.py` should pass

Injection Generator (2 days)

- **Owner:** Implement `ligo_gw/data/injection.py`

- BBH chirp waveforms (post-Newtonian approximation)
- Sine-Gaussian bursts
- SNR-normalized injection into noise
- Batch generation for training
- **Test:** `tests/test_data.py` should pass

Pull Request Process

1. Create feature branch: `git checkout -b feature/my-feature`
2. Make changes and commit
3. Run tests: `pytest tests/ -v`
4. Open PR with description linking to milestone
5. Code review before merge
6. Merge to main

Reporting Issues

Use GitHub Issues with label:

- `week-1` / `week-2` / etc. for milestone tracking
- `bug` / `enhancement` / `documentation`